

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号  
特開2000-347878  
(P2000-347878A)

(43)公開日 平成12年12月15日(2000.12.15)

(51)Int.Cl.<sup>7</sup>

G 0 6 F 9/45

識別記号

F I

G 0 6 F 9/44

テマコード\*(参考)

3 2 0 D 5 B 0 8 1

3 2 2 E

審査請求 有 請求項の数 6 O L (全 7 頁)

(21)出願番号 特願平11-159205

(22)出願日 平成11年 6 月 7 日(1999. 6. 7)

(71)出願人 000123262

沖縄日本電気ソフトウェア株式会社

沖縄県那覇市久米2丁目3番15号

(72)発明者 勢理客 智

沖縄県那覇市久米二丁目3番15号 沖縄日

本電気ソフトウェア株式会社内

(72)発明者 鶴田 洋

沖縄県那覇市久米二丁目3番15号 沖縄日

本電気ソフトウェア株式会社内

(74)代理人 100086645

弁理士 岩佐 義幸

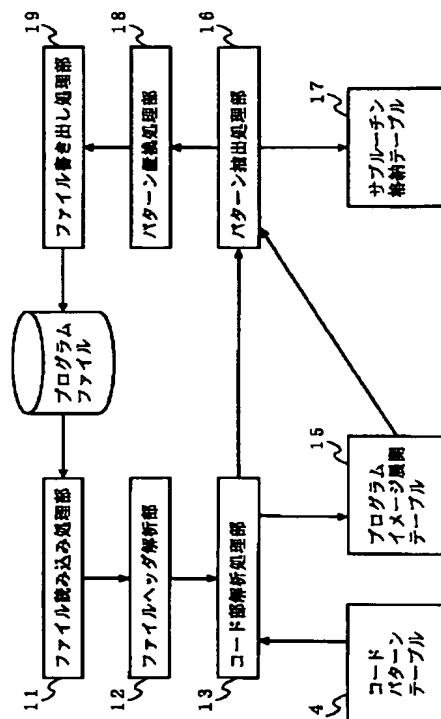
Fターム(参考) 5B081 CC22

(54)【発明の名称】 メモリ削減方法および装置

(57)【要約】

【課題】 ミスを生じさせず短時間でプログラム容量の削減ができるメモリ削減方法および装置を提供する。

【解決手段】 実装プログラムのプログラムサイズを削減するメモリ削減方法において、プログラム上の重複する処理を自動的に検索しサブルーチン化する。実行プログラムファイルを読み込んでその実行プログラムのコードを解析し、よりプログラムサイズが少なくなるようにプログラムコードを書き替え、別ファイルとして出力する。



## 【特許請求の範囲】

【請求項1】実装プログラムのプログラムサイズを削減するメモリ削減方法において、プログラム上の重複する処理を自動的に検索しサブルーチン化することを特徴とするメモリ削減方法。

【請求項2】実行プログラムファイルを読み込んでその実行プログラムのコードを解析し、よりプログラムサイズが少なくなるようにプログラムコードを書き替え、別ファイルとして出力することを特徴とする請求項1に記載のメモリ削減方法。

【請求項3】実行プログラムを読み出し、ワークエリアに読み込まれたファイルヘッダ部から、プログラム・スタート・アドレス及びプログラム・エンド・アドレスを取得し、プログラムコードデータをプログラムイメージ展開テーブルに展開する処理と、プログラムイメージ展開テーブルに展開されているレコードより検索パターンを生成し、プログラムイメージ展開テーブルの検索により同一のパターンが見つかった場合、その検索パターンをサブルーチン格納テーブルにエントリーナンバーと共に格納し、最後尾に復帰命令を付加する処理と、

全てのパターン検索が終わると、プログラムイメージ展開テーブルのエントリー格納エリアを検索し、CALL命令に必要な分を除き、プログラムイメージ展開テーブルに格納されているデータを前にシフトし、全てのシフトが終わると、プログラムイメージ展開テーブルの最後尾にサブルーチン格納テーブルに格納されているサブルーチンを転送する処理と、

変更されたプログラムイメージ展開テーブルを基に、プログラム・エンド・アドレスを算出し、変更したファイルヘッダ部をファイルに書き出し、プログラムイメージ展開テーブルのオペランド格納エリアに格納されている、変更後のプログラムイメージデータを書き出す処理とを有することを特徴とする請求項1または2に記載のメモリ削減方法。

【請求項4】複数のプログラムに跨って同一パターンのロジックを抜き出し、それぞれのプログラムで共通して使用可能にすることを特徴とする請求項1から3のいずれかに記載のメモリ削減装置。

【請求項5】実装プログラムのプログラムサイズを削減するメモリ削減装置において、実行プログラムファイルを読み込んでその実行プログラムのコードを解析し、よりプログラムサイズが少なくなるようにプログラムコードを書き替え、別ファイルとして出力することにより、プログラム上の重複する処理を自動的に検索しサブルーチン化することを特徴とするメモリ削減装置。

【請求項6】実行プログラムを読み出すファイル読み込み処理部と、ワークエリアに読み込まれたファイルヘッダ部から、プ

ログラム・スタート・アドレス及びプログラム・エンド・アドレスを取得するファイルヘッダ解析部と、オペランドのパターンを格納しているコードパターンテーブルに登録されているオペランド規則に従い、オペランドコード単位でプログラムイメージ展開テーブルへ転送し格納するコード部解析処理部と、

プログラムイメージ展開テーブルに展開されているレコードより検索パターンを生成し、プログラムイメージ展開テーブルの検索により同一のパターンが見つかった場合、その検索パターンをサブルーチン格納テーブルにエントリーナンバーと共に格納し、最後尾に復帰命令を付加するパターン抽出処理部と、

全てのパターン検索が終わると、プログラムイメージ展開テーブルのエントリー格納エリアを検索し、CALL命令に必要な分を除き、プログラムイメージ展開テーブルに格納されているデータを前にシフトし、全てのシフトが終わると、プログラムイメージ展開テーブルの最後尾にサブルーチン格納テーブルに格納されているサブルーチンを転送するパターン置換処理部と、

変更されたプログラムイメージ展開テーブルを基に、プログラム・エンド・アドレスを算出し、変更したファイルヘッダ部をファイルに書き出し、プログラムイメージ展開テーブルのオペランド格納エリアに格納されている、変更後のプログラムイメージデータを書き出すファイル書き出し処理部とを有することを特徴とする請求項5に記載のメモリ削減装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、メモリ削減方法および装置に関し、特に、ROM (Read Only Memory) 実装の実行プログラム作成におけるメモリ削減方法および装置に関する。

## 【0002】

【従来の技術】従来、ROM実装の実行プログラム作成において、ROMに入り切れないような場合が生じると、ソース見直しを行い、同様な処理を行っている個所はその個所をサブルーチン化して、容量の削減を行っていた。

## 【0003】

【発明が解決しようとする課題】しかしながら、従来の容量削減の場合、削減処理を機械的に行うことができない手で行うことから、対応者の技量によって結果が異なり一様な効果（削減容量）が得られないため、修正に時間を要しミスを生じさせる可能性があった。

【0004】この発明の目的は、ミスを生じさせず短時間でプログラム容量の削減ができるメモリ削減方法および装置を提供することである。

## 【0005】

【課題を解決するための手段】上記目的を達成するため、この発明に係るメモリ削減方法は、実装プログラム

10

20

30

40

50

のプログラムサイズを削減するメモリ削減方法において、プログラム上の重複する処理を自動的に検索しサブルーチン化することを特徴としている。

【0006】上記構成を有することにより、実装プログラムのプログラムサイズを削減するメモリ削減方法において、プログラム上の重複する処理が自動的に検索されサブルーチン化される。これにより、ミスを生じさせず短時間でプログラム容量の削減ができる。

【0007】また、この発明に係るメモリ削減装置により、上記メモリ削減方法を実現することができる。

【0008】

【発明の実施の形態】以下、この発明の実施の形態について図面を参照して説明する。

【0009】図1は、この発明の実施の形態に係るメモリ削減装置の構成を示すブロック図である。図1に示すように、メモリ削減装置10は、ファイル読み込み処理部11、ファイルヘッダ解析部12、コード部解析処理部13、コードパターンテーブル14、プログラムイメージ展開テーブル15、パターン抽出処理部16、サブルーチン格納テーブル17、パターン置換部18、及びファイル書き出し処理部19を有する。

【0010】ファイル読み込み処理部11は、指定されたファイルをファイルヘッダ部とプログラムコード部とに分けて、指定されたメモリに読み込む。ファイルヘッダ解析部12は、読み込まれたファイルヘッダ部から、プログラム・スタート・アドレス及びプログラム・エンド・アドレスを取り出す。

【0011】コード部解析処理部13は、ファイル読み込み処理部11で読み込まれたプログラムコード部を、オペランドのパターンを格納しているコードパターンテーブル14に登録されているオペランド規則に従い、オペランドコード単位でプログラムイメージ展開テーブル15へ転送し格納する。

【0012】なお、コード部解析処理部13においては、オペランド・コード解析処理過程でJMP命令やCALL命令等を検出した場合、その飛び先アドレスについても、プログラムイメージ展開テーブル15の飛び先格納エリアへ格納する。

【0013】図2は、コードパターンテーブルの説明図である。図2に示すように、コードパターンテーブル14は、オペランドとして成り立つ条件を基に、常に一定値をとる値のレンジを格納している固定データ長エリアと、必要データ長エリアと、一定値をとるオペランド格納エリアとを備える。図中のアセンブリイメージは、この発明を説明する上で擬似的に追加した情報である。

【0014】図3は、プログラムイメージ展開テーブルの説明図である。図3に示すように、プログラムイメージ展開テーブル15は、オペランド、元アドレス、飛び先アドレス、及びエントリーナンバーの各格納エリアを備える。

【0015】元アドレス格納エリアは、そのオペランドが元々存在していたアドレスを格納する。飛び先アドレス格納エリアは、JMP命令やCALL命令のように指定の番地に飛ぶような命令について、その飛び先のアドレスを格納する。エントリーナンバー格納エリアは、サブルーチン格納テーブルとの関係を示すために用いる。図中のアセンブリイメージは、この発明を説明する上で擬似的に追加した情報である。

【0016】パターン抽出処理部16は、プログラムイメージ展開テーブル15に格納されたオペランドコード単位のマシン語群の中から、5バイト以上のコードにより構成される同一パターンのコードを検索する。検索の結果、同一パターンがあった場合は、そのパターンをサブルーチン格納テーブル17に格納し、その最後尾に復帰命令(RET)を付加する。

【0017】また、そのパターンがあったプログラムイメージ展開テーブル15のエントリーナンバー格納エリアに、サブルーチン格納テーブル17のエントリーナンバーを格納する。

【0018】図4は、サブルーチン格納テーブルの説明図である。図4に示すように、サブルーチン格納テーブル17は、オペランド及びエントリーナンバーをそれぞれ格納するエリアを備えるワークエリアである。

【0019】パターン置換処理部18は、全てのパターン抽出の終了後、プログラムイメージ展開テーブル15のエントリーナンバー格納エリアを検索し、CALL命令に必要な分を除き、プログラムイメージ展開テーブル15に格納されているデータを前にシフトする。全てのシフトが終わると、プログラムイメージ展開テーブル15の最後尾にサブルーチン格納テーブル17に格納されているサブルーチンを転送し、シフトされた結果、飛び先が変更になったCALL、JMP命令を変更する。

【0020】ファイル書き出し処理部19は、パターン置換処理部18で変更されたプログラムイメージ展開テーブル15を基に、プログラム・エンド・アドレスを算出し、先ず、ファイルヘッダ部のプログラム・スタート・アドレス及びプログラム・エンド・アドレスを変更し、ファイルヘッダ部をファイルに書き出す。次に、プログラムイメージ展開テーブル15のオペランド格納エリアに格納されている、変更後のプログラムイメージデータを書き出す。これにより、実行プログラムのメモリサイズを削減可能にする。

【0021】次に、図1のメモリ削減装置10の動作を説明する。図5は、図1のメモリ削減装置の動作を説明するフローチャートであり、図6は、シフト後のプログラム・イメージ展開テーブルの説明図である。

【0022】図5に示すように、先ず、ファイル読み込み処理部11は、実行プログラムを読み出し、ファイルヘッダ部とプログラムコード部とに分けて、ワークエリアに読み込む(ステップS101)。読み込まれたファイ

ルヘッダ部から、ファイルヘッダ解析部12を使用して、プログラム・スタート・アドレス及びプログラム・エンド・アドレスを取得する(ステップS102)。

【0023】ステップS101において、読み込んだプログラムコード部の先頭が“E910005056…”と仮定し説明すると、コード部解析処理部13では、先ず、プログラムイメージ展開テーブル15を値0で初期化し、次に、プログラムコードデータの先頭から、コードパターンテーブル(図2参照)と比較しながらプログラムイメージ展開テーブル15に展開していく。ここで

は、先頭の“E9”で始まるものを、コードパターンテーブルから検索する。

【0024】検索の結果、先頭レコードのものが検出され、そのレコードの固定データ長エリアが1で必要データ長エリアが3であるため、“E91000”が意味のあるオペランドとして、プログラムイメージ展開テーブル15(図3参照)の先頭レコードに格納される。

【0025】引き続き、“5056”に対しコードパターンテーブル(図2参照)を基に検索が行われ、“50”、“56”がそれぞれ別レコードとしてプログラムイメージ展開テーブル15に格納される。この処理を、プログラム・エンド・アドレスまで繰り返し、プログラムコードデータ全てをプログラムイメージ展開テーブル15に展開する(ステップS103)。

【0026】次に、プログラムイメージ展開テーブル15に展開されているレコードより、5バイト以上になる単位で検索するパターンを生成する(ステップS104)。検索パターンの生成が成功したか否かを判断し(ステップS105)、生成に成功した(Yes)場合、そのパターンが他に存在しないかプログラムイメージ展開テーブル15を検索する(ステップS106)。

【0027】検索により、同一のパターンがあるかを判断し(ステップS107)、同一のパターンが見つかった(Yes)場合、プログラムイメージ展開テーブル15の該当レコードのエントリーナンバー格納エリアに、エントリーナンバーを格納する(ステップS109)。その後、ステップS104へ戻る。

【0028】このときのエントリーナンバーは、同一パターン検索の1番目のパターンであれば01をセットし、n番目のパターンであればnをセットする。その際、その検索パターンをサブルーチン格納テーブル(図4参照)にエントリーナンバーと共に格納し、最後尾に復帰命令を付加する(ステップS108)。

【0029】以下、ステップS104からステップS109迄の流れを具体的に説明する。プログラムイメージ展開テーブル15が図3に示すような場合、①と②が同一パターンと見なされ、この部分が、図4の①にあるようにサブルーチン格納テーブルに格納され、復帰命令(RET)が付加されることになる。

【0030】次に、全てのパターン検索が終わると、プ

ログラムイメージ展開テーブル15のエントリー格納エリアの値が1以上のものについて、そのレコードより上位3バイトのみを残し1レコードとして格納し、残りは前にシフトする(ステップS110)。全てのシフトが完了すると、プログラムイメージ展開テーブル15の最後尾に、ステップS108で格納したサブルーチン群を転送する(ステップS111)。

【0031】次に、ステップS110において、プログラムイメージ展開テーブル15に3バイト確保したオペランド格納エリアのコードを、CALL命令に変更し、そのエントリーナンバーに対応するサブルーチン(ステップS111において最後尾に転送したサブルーチン)を呼び出すように変更する(ステップS112)。図6の①の場合は、エントリー01番のサブルーチンを呼び出すように変更する。

【0032】続いて、ステップS112までで生成されたプログラムイメージ展開テーブル15の最終アドレスを取得し、先ず、ステップS101において読み込んだファイルヘッダ部のプログラム・エンドアドレスを変更し書き出す。次に、プログラムイメージ展開テーブル15のオペランド格納エリアに格納されている変更後の情報を、追加の形で書き出す。これにより、プログラムサイズを圧縮した新たな実行プログラムが生成される。

【0033】このように、この発明によれば、実行プログラムファイルを読み込み、その実行プログラムのコードを解析し、よりプログラムサイズが少なくなるようにプログラムコードを書き替えて、別ファイルとして出力することができる。

【0034】これにより、同一パターンの処理を行うようなものについてサブルーチン化が自動的に行えるため、ROM実装プログラム等のように、プログラムサイズ削減が厳しく要求されるような場合において、ミスを生じさせず短時間でプログラム容量の削減ができる。

【0035】なお、上記実施の形態においては、一つの実行プログラムに対するメモリ削減方法について述べたが、複数のプログラムに跨って同一パターンのロジックを抜き出し、それぞれのプログラムで共通して使用可能にしてもよい。これにより、システム全体としてのメモリ削減が可能となる。

【0036】

【発明の効果】以上説明したように、この発明によれば、実装プログラムのプログラムサイズを削減するメモリ削減方法において、プログラム上の重複する処理が自動的に検索されサブルーチン化されるので、ミスを生じさせず短時間でプログラム容量の削減ができる。

【0037】また、この発明に係るメモリ削減装置により、上記メモリ削減方法を実現することができる。

【図面の簡単な説明】

【図1】この発明の実施の形態に係るメモリ削減装置の構成を示すブロック図である。

【図2】コードパターンテーブルの説明図である。

【図3】プログラムイメージ展開テーブルの説明図である。

【図4】サブルーチン格納テーブルの説明図である。

【図5】図1のメモリ削減装置の動作を説明するフローチャートである。

【図6】シフト後のプログラム・イメージ展開テーブルの説明図である。

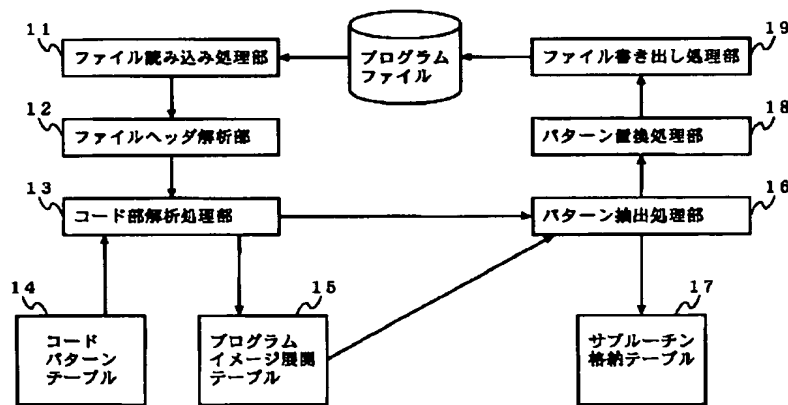
【符号の説明】

10 メモリ削減方法および装置

- \* 11 ファイル読み込み処理部
- 12 ファイルヘッダ解析部
- 13 コード部解析処理部
- 14 コードパターンテーブル
- 15 プログラムイメージ展開テーブル
- 16 パターン抽出処理部
- 17 サブルーチン格納テーブル
- 18 パターン置換部
- 19 ファイル書き出し処理部

\*10

【図1】



【図2】

必要データ長		固定データ長		オペランド格納エリア		アセンブラ・イメージ	
08	01	E9				JMP XXXX	
08	01	E8				CALL XXXX	
01	01	60				PUSH AX	
01	01	58				POP AX	
02	01	74				JZ XXXX	
01	01	1E				PUSH DS	
08	01	B8				MOV DS, 0F0F	
01	01	1F				POP DS	
01	01	5E				POP SI	
01	01	56				PUSH SI	
01	01	1E				PUSH DS	
02	02	8ED8				MOV DS, AX	
01	01	90				NOP	

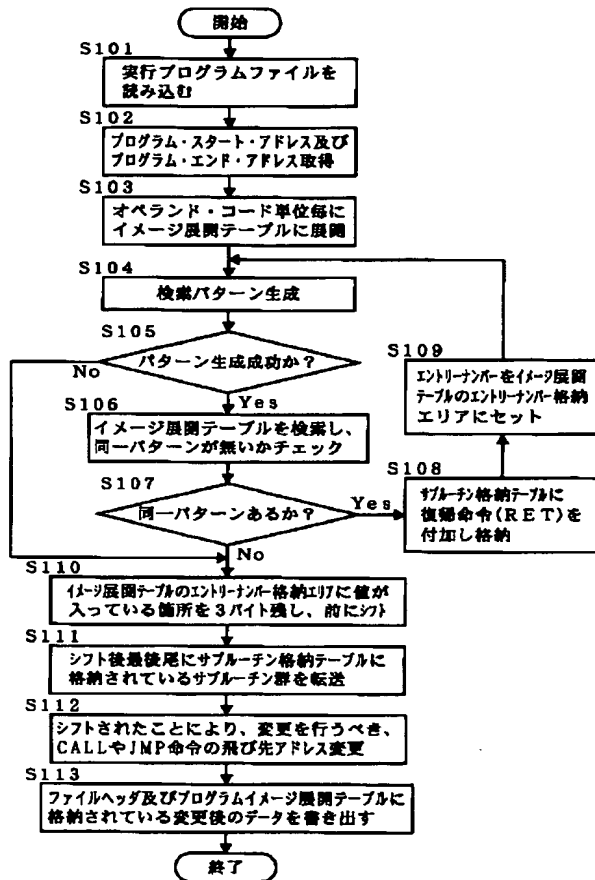
【図3】

エントリーナンバー格納エリア		オペランド格納エリア		飛び先アドレス		元アドレス		アセンブラ・イメージ	
00		E91000	0010	0000				JMP 0010	
00		E8FD08	0500	0100				CALL 0500	
00		60		01FF				PUSH AX	
01	56			0200				PUSH SI	
01	1E			0201				PUSH DS	
01	E8FF0F			0202				MOV DS, 0F0F	
01	8ED8			0205				MOV DS, AX	
01	1F			0207				POP DS	
01	5E			0208				POP SI	
00	56			0209				POP AX	
00	90			02FF				NOP	
01	56			0300				PUSH SI	
01	1E			0301				PUSH DS	
01	E8FF0F			0302				MOV DS, 0F0F	
01	8ED8			0305				MOV DS, AX	
01	1F			0307				POP DS	
01	5E			0308				POP SI	
00	90			0309				NOP	

【図4】

エントリーナンバー格納エリア		オペランド格納エリア		アセンブラ・イメージ	
01	50			PUSH AX	
01	56			PUSH SI	
01	1E			PUSH DS	
01	B83F0F			MOV DS,0F3F	
01	8ED8			MOV DS,AX	
01	1F			POP DS	
01	6E			POP SI	
01	68			POP AX	
01	03			RET	
02	50			PUSH DS	
02	56			PUSH SI	

【図5】



【図6】

エントリーナンバー格納エリア

オペランド格納エリア

飛び先アドレス

元アドレス

アセンブラ・イメージ

00	E91000	0010	0000	JMP 0010	
00	E9FD03	0500	0100	CALL 0500	
00	50		01FF	PUSH AX	
01	88XXXX		0200	CALL 17'h-ジャンプ-01	①
00	58		0209	POP AX	
00	90		02FF	NOP	
01	88XXXX		0300	CALL 17'h-ジャンプ-01	②
00	90		0300	NOP	
00	56			PUSH SI	
00	1E			PUSH DS	
00	B83F0F			MOV DS,0F3F	
00	BED8			MOV DS,AX	
00	1F			POP DS	
00	5E			POP SI	
00	C3			RET	